UNIVERSITAT DE BARCELONA

Treball final de grau

GRAU D'ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

Where Am I Eating? Image-based Food Menu Recognition

Autor: Marc Valdivia Arriaza

Director: Marc Bolaños Solà
Realitzat a: Departament de Matemàtiques i Informàtica

Barcelona, June 27, 2018

Abstract

Food has become a very important aspect of our social activities. Since social networks and websites like Yelp appeared, their users have started uploading photos of their meals to the Internet. This factor leads to the development of food analysis models and food recognition.

We propose a model to recognize the meal appearing in a picture from a list of menu items (candidates dishes). Which could serve for the recognize the selected meal in a restaurant. The system presented in this thesis does not need to train a new model for every new restaurant in a real case scenario. It learns to identify the components of an image and the relationship that they have with the name of the meal.

The system introduced in this work computes the similarity between an image and a text sequence, which represents the name of the dish. The pictures are encoded using a combination of Convolutional Neural Networks to reduce the input image. While, the text is converted to a single vector applying a Long Short Term Memory network. These two vectors are compared and optimized using a similarity function. The similarity-based output is then used as a ranking algorithm for finding the most probable item in a menu list.

According to the Ranking Loss metric, the results obtained by the model improve the baseline by a 15%.

Abstract - Catalan

El menjar s'ha convertit en un aspecte molt important a la nostra vida social. L'aparició de les xarxes socials i de pàgines com Yelp ha provocat que els seus usuaris comencin publicar fotografies del seu àpat a Internet. Aquest fet ha liderat el desenvolupament d'aplicacions d'anàlisis i reconeixement de menjar.

Nosaltres proposem un model capaç de reconèixer el plat de menjar que apareix en una imatge en relació al menú d'un restaurant. El sistema que presentem no necessita entrenar-se de nou per a tots els menús o restaurants. El que fa és aprendre a identificar els components de la imatge i relacionar-los amb els noms dels plats.

El funcionament del model es basa a calcular la similitud entre la imatge i una seqüència de text, que representarà cadascun dels elements de la llista (menú). Les imatges són codificades fent servir una combinació de xarxes convolucionals per reduir l'element d'entrada. El text, per altra banda, és transformat a un únic vector per mitjà d'una xarxa Long Short Term Memory. Aquests dos vectors són comparats i optimitzats el seu resultat fent servir una funció de similitud. Aquesta sortida, basada en la similitud, és far servir per crear un rànquing dels noms més probables en relació a la imatge donada.

En relació a la mètrica de Ranking Loss, els resultats obtinguts pel model milloren l'error bàsic en un 15%.

Abstract - Spanish

La comida se ha convertido en un aspecto muy importante en nuestra vida social. La aparición de redes sociales y páginas web como Yelp han provocado que sus usuarios empiecen a subir fotografías de sus comidas a Internet. Este hecho ha liderado el desarrollo de aplicaciones de análisis y reconocimiento de comida.

Nosotros proponemos un modelo capaz de reconocer un plato de comida representado en una fotografía con relación al menú de un restaurante. El sistema que presentamos no necesita entrenarse de nuevo para todos los menús o restaurantes. Aprende a identificar los componentes de la imagen y relacionarlos con los nombres de los platos de comida.

El funcionamiento del modelo se basa en calcular la similitud entre la imagen y una secuencia de texto, que representa los elementos de la lista (menú). Las imágenes son codificadas usando una combinación de redes convolucionales para reducir el elemento de entrada. El texto, por otra banda, se transforma en un único vector usando una red Long Short Term Memory. Estos dos vectores se comparan usando una función de similitud. La salida, basada en la similitud, se utiliza para crear un ránking de los nombres más probables en relación con la imagen de entrada.

En relación con la métrica de Ranking Loss, los resultados obtenidos por el modelo mejoran el error básico en un 15%.

Acknowledgments

I would first like to thank my thesis advisor Marc Bolaños Solà for his patience guiding me through this project and advices to constantly improve my work. The door of his office was always open whenever I needed or had a question about my research or writing.

I would also like to thank my coworkers from NPAW (Nice People At Work) for their help and support. Andrea Indave and Patrycja Pypczynski for helping me with all the writing, their help saved me in many occasions. Felipe Quirce and Iosu Miral (eDreams) for their technical advice and suggestions to improve my results. And anybody at the office who suffered me during the last days of the deadline.

Finally, I must express my very profound gratitude to my parents and friends for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Food is one of the key factors in peoples lives. Nowadays, food does not only cover a basic need, but it has become a really important aspect of our social activities. Since social network systems appeared and, with them, food-focused applications (like TripAdvisor, Yelp, etc.) their users have started uploading photos of their meals to the Internet. It seems to be a strong and visible tendency in todays society to share pictures of absolutely every piece of food that we taste; exotic or local, fancy-looking or ordinary. Moreover, people post, on many different social media channels, plenty of videos of all visited food places. Every single day, thousands of people use social media to make recommendations, promote a particular place or give their friends a warning about a nearby restaurant. That is why, tags and location opportunities were introduced for all social media users to make their posts easier and faster to create.

The purpose of this thesis work is to create a predictive model that can determine the similarity between a food image taken in a restaurant to their corresponding menu item. The proposed methodology does not need to train a new model for each restaurant, it will learn to understand meal names in relation to a set of examples in a language model. We should point out the difficulty of the problem because of the

context where we are working. Restaurants usually use fancy names to refer to the dishes just to get the attention of their customers. Additionally, food presentation is different in every restaurant, having a high intra-class variability. Chefs try to hide the ingredients using colorful plates and sauces.

1.1 Health and Leisure

The book Food and Health in Europe [19] introduces the relationship that exists between food consumption and people's health. In Europe, despite being a first-world region, more than 4 million people die each year due to chronic diseases linked to unhealthy lifestyles. These people have a high probability of suffering from strong shortage of daily physical activity and regular consumption of food that has high levels of fat. In many of these cases, the lack of basic knowledge is a crucial factor in all problems: a majority of people simply do not pay much attention to their eating habits. Moving our focus from the European society to the American one, it is important to mention that the above-discussed numbers are even worse. It is mentioned in the article [21] that a great number of deaths related to coronary heart diseases are caused by a group of major risk factors among which bad food habits are at the top.

On the other hand, it is a fact for a lot of people that being healthy is considered trendy nowadays. Thanks to social networks, and influencers among others, who share their healthy lifestyle in the social media channels on a daily basis, the importance of dropping out of an unhealthy way of life is gaining more and more fans. That is why social media plays a significant role in convincing people to change their

harmful eating and life habits.

Nowadays going out for dinner and enjoying a cosy atmosphere in a restaurant is not enough. People feel the urge to post on their social networks not only where they are going but also what they are about to eat. The healthier (and better looking) your food is, the better. Because of this important fact, todays restaurants are really visible on-line and they tend to use many different Internet channels to remain in the center of their customers attention. They want to be tagged in the pictures posted by their clients and get positive reviews from them that can be shown to the great audience on their web pages. Social networks sometimes seem not to be enough to collect and present all user experience, that is why there is a considerable movement to arise food-based applications like Yelp, that help their users get what they want beforehand. Many clients want to know in advance what the quality of the service is in the place they plan to visit. A great amount of people prefer reviewing different users opinions before visiting a particular restaurant. Having the chance to take a look at the food that they will find in the restaurant is also attractive to food lovers. All those facts help to understand and illustrate the rising popularity of food-based applications worldwide.

1.2 Food Analysis and Deep Learning

Considering the huge number of pictures of meals that people upload to the Internet, food analysis has become popular in the Deep Learning field. These photos usually have a description or personal evaluation of the food eaten. They are generating a big amount of information, that is what deep learning needs. That is the reason why have appeared public datasets. They are available to the data scientist to create their own predictive systems. Amongst others, the Food-101 [7] is a well-known

dataset for food analysis. That is why these problems cause a great interest in the scientific community. Because of that, multiple applications in the real world could be created to make our life easier. Computer vision can be performed using deep learning techniques. Because of a high percentage of the food related information are images, that field is the most exploded for this kind of problems.

The model that we present in this thesis combines different deep learning techniques to perform an image-based food analysis, we combine the pictures of the meals taken by the users and name each one of the dishes.

Food recognition is one of the most popular problems nowadays. It is the machines capacity to recognize a particular piece of food presented in a picture. The systems prediction will be selected from a limited number of food types, also known as classes. This type of applications can help us create a useful diary of our food habits. The scientific community is not only working in food recognition, there are other applications of food analysis that are very useful in peoples lives. Food localization systems detect multiple meals in picture [6]. Additionally can use the GPS information of the devices to determine the place where the user is eating. Calories estimation applications and ingredients detections keep an automatic diary of food consumption, helping people with alimentary disorders to be healthier.

The food recognition use to be separated in two problems. The recognition process by itself, occurs when you provide a group of pixels representing a single food type to the system and tries to determine the class to which it belongs [1]. The recognition process preceded by detection. The images do not usually contain one meal presented, for example in a restaurant. In these cases we need to run a food localization algorithm to know the different clusters of pixels that contain the possible meals

in the picture. This combination of processes is common in self-service restaurants [2].

1.3 Restaurant Food Recognition

As it was said in the beginning of this chapter, there are applications focused on understanding customers experience with food. The main goal of those applications is to provide useful opinions about the restaurants and the food they serve. These sites have plenty of information but they are not able to classify a picture in the restaurants menu automatically. It is the user who should do this process manually. That is the reason why predictive models appeared to solve this specific problem: locate the restaurant where the customers are eating and recognize the meal that they chose from the menu [28].

In this chapter we highlight two types of problems that are currently on the scope of deep learning, but we want to go further. The novelties of our work are the following:

- Collecting our own dataset from the Yelp [1] website. We decided to create the dataset at our own to face a real problem, instead of using the ones provided by the scientific community. Summarizing the information appearing in it, the dataset contains the dishes and image examples of them for each one of the restaurants in the set of data. In order to collect the dataset, we used a web scraper that we developed.

- Proposing a model to determine the similarity between a food picture and the dish description provided in the restaurants menu. Thanks to the knowledge

[1] http://www.yelp.com

model learned, the system should be able to find the similarity for any tuple of image and text, including never seen examples.

- We propose the first model for food menu recognition for any restaurant. The system does not need previous information of a specific restaurant or any set of examples for a specific class to perform the prediction.

- The results obtained over the collected data improve the baseline by a 15%.

This document is organized in 7 chapters. The introduction is the current one, where we present the problem and the context where we are working. In the related work we explain previous papers published in relation with the problem that we want to solve. We compare their proposals with ours and their advantages and disadvantages. The basic theoretical concepts, as well as our proposed model are introduced in the methodology chapter, where we explain the deep learning models used to build our system. The dataset section introduces the data used to train our model and how it was collected. In the results, we explain the set of experiments done to choose the best parameters of the proposed model and their performance. The last two chapters are the discussion and conclusions, where we show the outcome of the work, the benefits to the scientific community and the future work.

Chapter 2

Related Work

In this section we present previous work done in the field of food analysis. We cover the food detection and recognition in one section and the application of these techniques in a restaurant context, where the system should select the menu item that the customer has chosen.

2.1 Food Analysis

Food analysis has the main objective of improving people's lives. Despite of the purpose of this thesis is focused in social activities, we would like to introduce some works related to the first topic. In the paper [27] the authors present a mobile phone-based calories monitoring system to track the calories consumption for people with special nutritional needs. Focused on diabetes, the publication Automated food ontology construction mechanism for diabetes diet care [14] estimates the amount of carbohydrate present in a meal from an image. The book [15] shows image-based food recognition model to create a calories diary intake, that is a key factor in weight loss. The proposed system not only recognizes the food category but the portion

size too. Food analysis also helps to detect bacteria in the food: direct detection of trimethylamine in meat-based products using ion mobility spectrometry [8] proposes a system to recognize the degradation in meat products. This same issue is faced by The fish is bad [4], but from other perspective; they introduce sensors to classify the odor of the fish, saying if it is in a good or bad state.

The field of food detection and recognition has been worked from different perspectives. The paper Simultaneous food localization and recognition [6] introduces the use of egocentric images to perform food detection and recognition. The proposed model used an activation map to detect the different dishes appearing in the picture and then recognize each one of the food types present in the bounding boxes. Other perspective is presented in the publication Food Ingredients Recognition through Multi-label Learning [5] which uses a state of the art CNN to predict a list of ingredients appearing in the meal, even the recipe has never been seen by the system.

2.2 Multimodal Food Analysis

Food analysis uses context or additional information to improve the accuracy of the predictions. This complementary data usually is not of the same type (i.e. images and text, video, sounds, etc.). Multimodal Deep Learning [18] solves this particular problem, learning features over multiple modalities. in the paper [29] the authors use this kind of model to relate the image rankings in the queries results with the click features.

The paper Learning Cross-modal Embeddings for Cooking Recipes and Food Images [22] introduces a new large-scale dataset with more than 800.000 images and

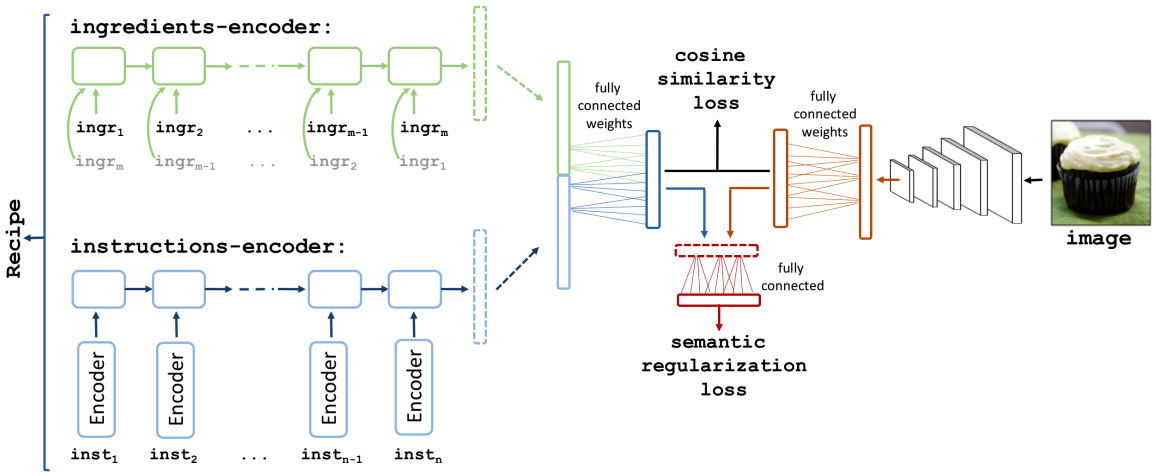

Figure 2.1: Learning Cross-modal Embeddings for Cooking Recipes and Food Images model topology.

1.000 recipes. The predictive model presented in the paper tries to join images and recipes through a retrieval task. The proposed solution generates two vectors. One of the vectors represents the image and the other represents the recipe. The paper explains how the dataset was built or the word embedding used, but the following explanation will focus on the model. We are interested in how dimensionality of the images and the recipes could be reduced to be comparable. Fig. 2.1 shows the topology of the model presented in this paper. A CNN network transforms the image to a single vector summarizing the information of the picture. At the same time, two different LSTM networks process the recipes ingredients and steps to combine their output in a single vector. At this point, the image and the recipes are represented as two different vectors, to be comparable they need to have the same size. This issue is solved by applying a fully connected layer of the same size to each one of the vectors. Finally, the cosine similarity loss determines if the recipe represents the food at the image.

This work is very similar to ours. The problem that we want to solve also has two different inputs. We need to compare an image and a text sequence, so it is an image retrieval problem like this. The main difference of our proposal is that, instead of

using a general purpose CNN to generate the features vector of the image, the system that we used has been trained with food related images.

2.3 Restaurant Food Recognition

Restaurants and food delivery companies are interested in systems of menu recognition to create more efficient payments processes. In the paper [3] the authors propose an automatic food journaling using our smartphones. They use state of the art computer vision techniques and add context information of the restaurant to predict the food being consumed by the costumer. The publication [27] creates a calorie estimation from web video cameras in fast food restaurants across the United States. They focused on a reduced group of restaurants to understand the obesity problem.

The paper Geolocalized Modeling for Dish Recognition [28] introduces the context of the pictures to recognize the dish appearing in the image. Using the GPS information provided by the smart-phones they can determine a set of restaurants where the picture has been taken. This reduces the search space, which is really important when you try to determine the restaurant and menu item that appeared in the picture taken by the user. The system needs to train a discriminative model for each pair of restaurants in the dataset comparing their menus and images. The complexity of the problem is huge if we try to perform a one vs all algorithm. They use the restaurants context information to train a model only for pairs of restaurants which are close enough. Fig. 2.2 shows an schema of the model, where we can appreciate that the models are geolocalized. This means that the algorithm applies the trained model based on the GPS information of the input.

The problem of the model presented in this paper is the need to train a new model

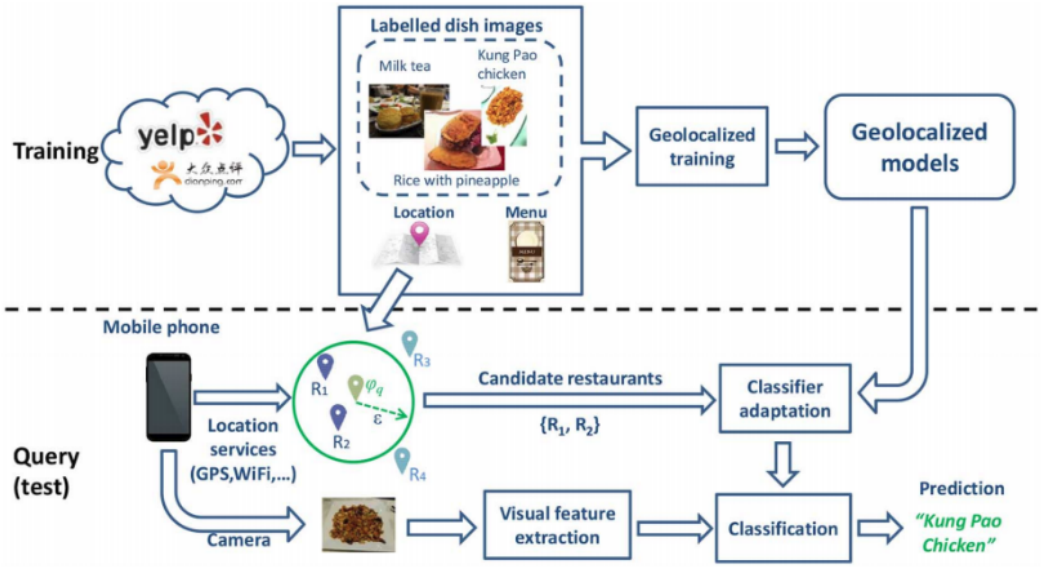

Figure 2.2: Geolocalized Modeling for Dish Recognition model topology.

when a new restaurant is added to the dataset. Additionally, the system only works if the restaurant appears in the collected data. Nevertheless, the predictive model that we will present is built to work with previously unseen data. It does not need to have image examples of a restaurant dish to perform the prediction.

The paper Siamese Recurrent Architectures for Learning Sentence Similarity [17] is an adaption of the Long-Short Term Memory Network. The purpose of the model is to find semantic similarity between sentences. The system gets two inputs, represented as two text sequences. The output is a single value between 0 and 1 indicating the similitude of the inputs. Fig 2.3 is a representation of the model. The similarity value is computed using the exponential negative Manhattan distance, represented in the fig. 2.1. Additionally, the optimization function is the contrastive loss. It was introduced by Yann LeCun at the paper Dimensionality Reduction by Learning an Invariant Mapping [11].

$$g(h_{Ta}, h_{Tb}) = exp(-\|h_{Ta} - h_{Tb}\|_1) \exists [0, 1] \tag{2.1}$$

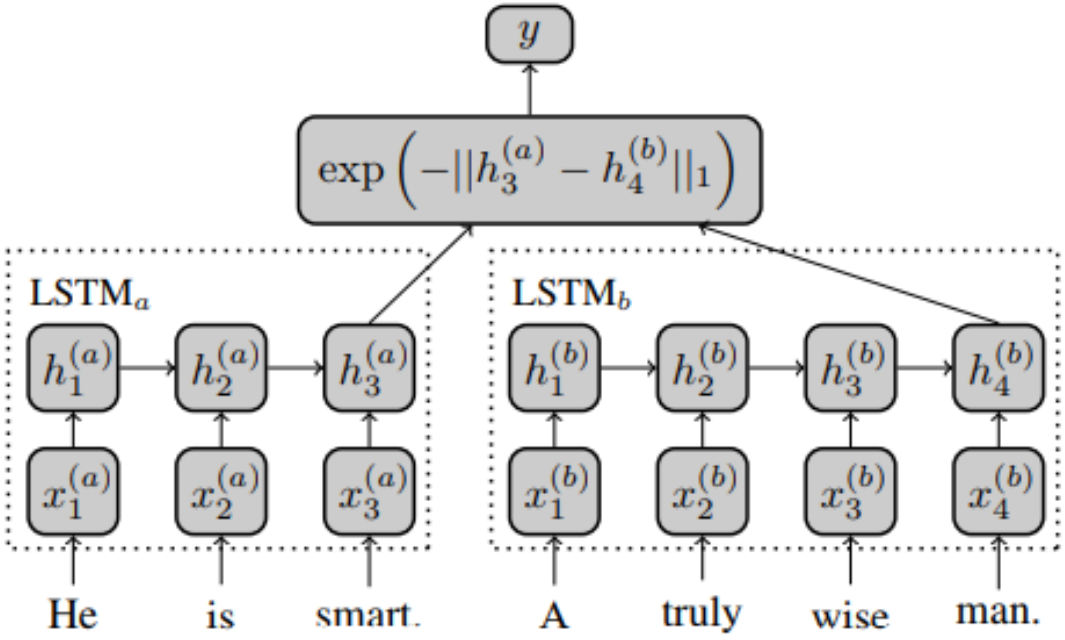

Figure 2.3: Siamese Recurrent Architectures for Learning Sentence Similarity model topology.

The work done in this paper is really similar to our problem. The difference is that, instead of comparing two sentences, we have to find the similarity between an image and a text sequence.

The problem that is present in food recognition is that one is limited to a number of classes. This means that if the model was not trained to recognize some type of food (i.e. Mexican, Indian, etc.), it will never provide it as a possible output.
On the other hand, the complexity in the restaurants food recognition resides in the need of training a different model for each restaurant. These models could be very accurate, but the number of outputs is also limited to the restaurants menu. The way proposed in this paper to help resolve this issue is completely different and more complex. Our model tries to learn all possible names associated to the same dish, depending on the restaurant where it is served. What is more, our algorithm should be able to take a completely new restaurants menu (never seen before) and a totally

new picture associated to one of the menus items and find out the correct choice. This means that it is not needed to train a new model for each meal of the restaurant because the network will actually learn itself how to read a menu and identify each meal that appears on it.

Chapter 3

Methodology

Throughout this chapter we will first introduce the different components (different types of neural networks) used to build our system. The predictive model that we have built has to receive completely different kinds of inputs. On the one hand, we will receive an image representing the meal to recognize. On the other hand, a text sequence that will be the foods menu item to compare.
The topics that we will explain in the following sections are:

1. Neural Networks
2. Convolutional Neural Networks
3. Word Embedding
4. Recurrent Neural Networks & LSTM

Some topics listed above are not easy to learn, plus they treat a wide range of different problems, so the explanations throughout this chapter will be focused on the objectives of each model and on giving a preview of how they are built.

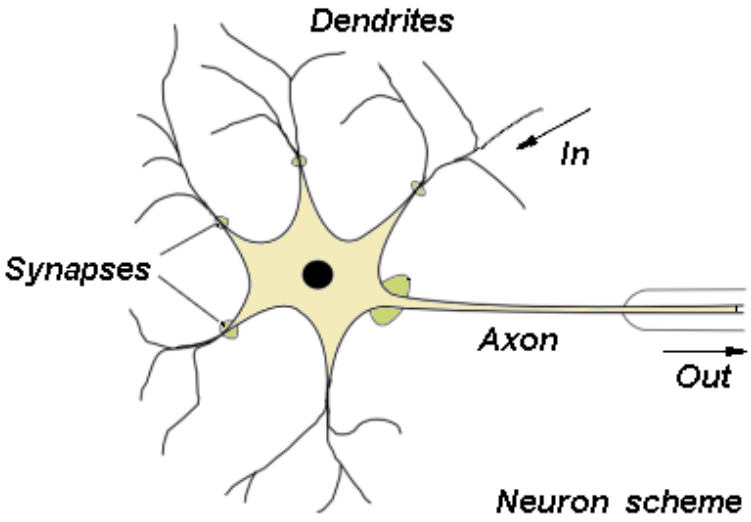

Figure 3.1: Simple representation of a human neuron.

3.1 Neural Networks

In this first section we introduce the concept of neural network from a basic perspective, a reader interested in the field can check the post Introduction to Neural Networks [1].

Neural networks try to emulate the human brain, which is considered the best known learning structure. The figure 3.1 is a simplified version of a human neuron, which is composed by multiples inputs and a single output. Neurons dont treat each input equally, some of them are designed to learn specific patterns, giving more importance to some inputs than others. The neurons that are designed to recognize a specific topic give a higher activation value to the related incoming information than the others.

The behavior explained above is the one that artificial neurons try to simulate by using the schema shown in the fig. 3.2. The multiple inputs are regularized by some

[1] http://home.agh.edu.pl/ vlsi/AI/intro/

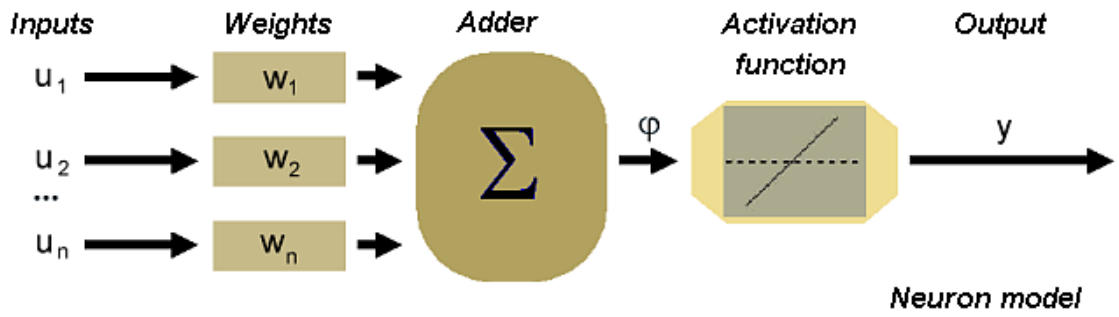

Figure 3.2: Artificial neuron trying to imitate the human neuron behavior.

weights: the most important inputs have higher weights. The next step combines the partial results in one single output, performing an addition of all the responses. Finally, the neuron applies an activation function to shape the result in the most appropriate way.

The standard formula applied to the artificial neurons is called perceptron and is described by:

$$\sum_{i}^{n} x_i w_i + b \tag{3.1}$$

The result of the previous formula will be the input of the activation function. Some well-known activation functions are the linear, sigmoid and threshold, but they are applied to the basic neural network structures. For example, the most used function in the CNNs is the rectifier linear unit (ReLu) and in the recurrent networks they are the sigmoid and tanh functions. The linear activation function (Fig. 3.2) could retrieve any result between minus infinite and infinite. The results of the sigmoid function (Fig. 3.3) are between 0 and 1. Finally, the ones given by the threshold function (Fig. 3.4) could be just 0 or 1. Fig. 3.3 shows the behavior of these three activation functions. The rectified linear unit (ReLu) is a modification of the linear function and is used by a lot of networks. The result of the ReLu is just the maximum between 0 and the output of the linear activation function, shown at the equation 3.5.

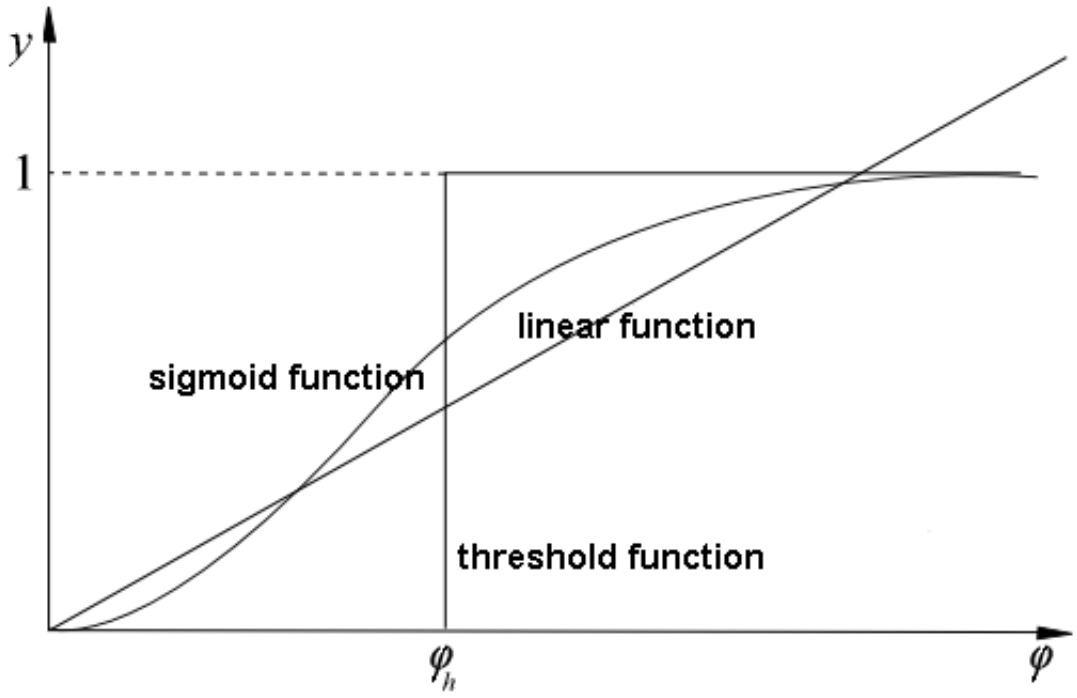

Figure 3.3: Linear, Sigmoid and Threshold activation functions outputs.

$$y = k\varphi \tag{3.2}$$

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \tag{3.3}$$

$$y = \left\{ \begin{array}{l} 1 \text{ for } \varphi > \varphi_t \\ 0 \text{ for others} \end{array} \right\} \tag{3.4}$$

$$R(z) = max(0, z) \tag{3.5}$$

We have just seen the options applicable to a single neuron, but the power of neural networks resides in the combination of multiple neurons distributed in different layers creating complex nets. The fully connected or dense layer (FC) is the most popular one, where all the inputs are connected to each single neuron of the next layer. The

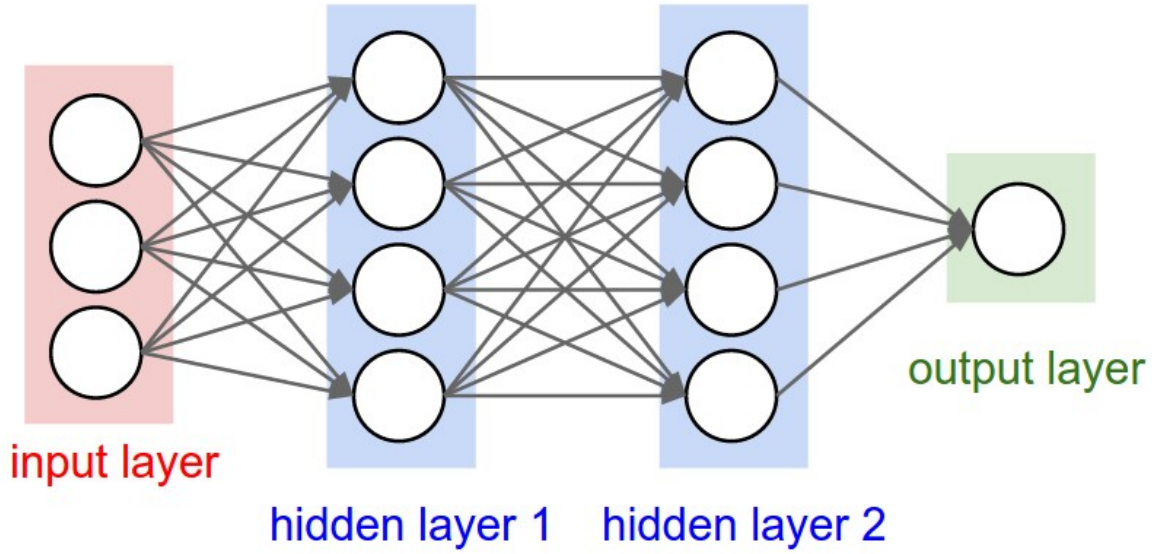

Figure 3.4: Fully connected network with two hidden layers.

inputs of a hidden layer are the outputs of the previous one. Fig. 3.4 shows an example of a neural network with 2 fully connected hidden layers.

Once the neural network is built, the system needs to train the model and find the most appropriate weights for each neuron. These weights have to minimize the error of the output. This process, where the network tries to fine-tuning its parameters, is called back-propagation [13]. The name is given because of the fact that the optimization process begins at the output of the model and propagates the error, given by the derivative, from the back to the top.

3.2 Convolutional Neural Networks

Convolutional Neural Networks, or CNNs, are similar to the basic neural networks seen in the previous section. This kind of systems are specifically designed to work with images, the hidden layers are replaced by convolutional masks to reduce the networks parameters number. Additionally, they are useful to find pixels correlation

and patterns in the images of the same topics. CNNs are powerful machine learning algorithms to perform image recognition and the process could be divided in three basics steps, which is not mandatory to do in the presented order. The following lines introduce the basic pipeline in a convolutional networks, but there are other advanced procedures to improve the accuracy of the models (i.e. dropout, batch normalization, etc.) that we do not explain here.

1. Convolution
2. Pooling
3. Classification

The convolution process, shown in the fig. 3.5, applies multiples masks to the images matrix with the purpose of extracting the key features of the picture. It preserves the spatial relationships of the pixels. The fig. 3.6 shows max-pooling 2x2 process the pooling step, which reduce the size of the image by half. The pooling is used to reduce the dimensionality of the inputs (by a given factor) and reduces the computational time. The first layers of the networks detect colors and edges, while the last ones recognize objects and relations between them. Finally, the classification consists in creating a fully connected layer with as many neurons as classes to classify. Fig. 3.7 shows a summary of the whole process, from the input image to the classification output.

The ones interested in the field of computer vision applying CNNs could read the article written in the Cambridge Code Academy website, Deep learning for complete beginners: convolutional neural networks with keras [2].

[2]https://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html

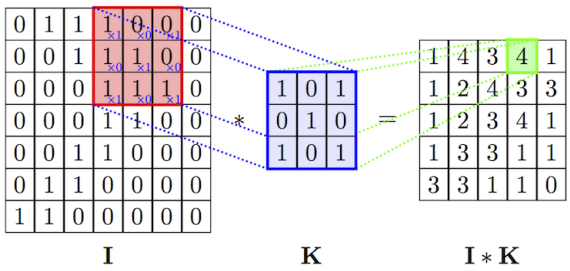

Figure 3.5: Convolution process of a 3x3 mask.

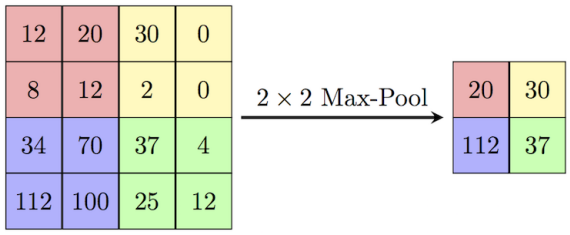

Figure 3.6: 2x2 Max-pooling reducing the size by half.

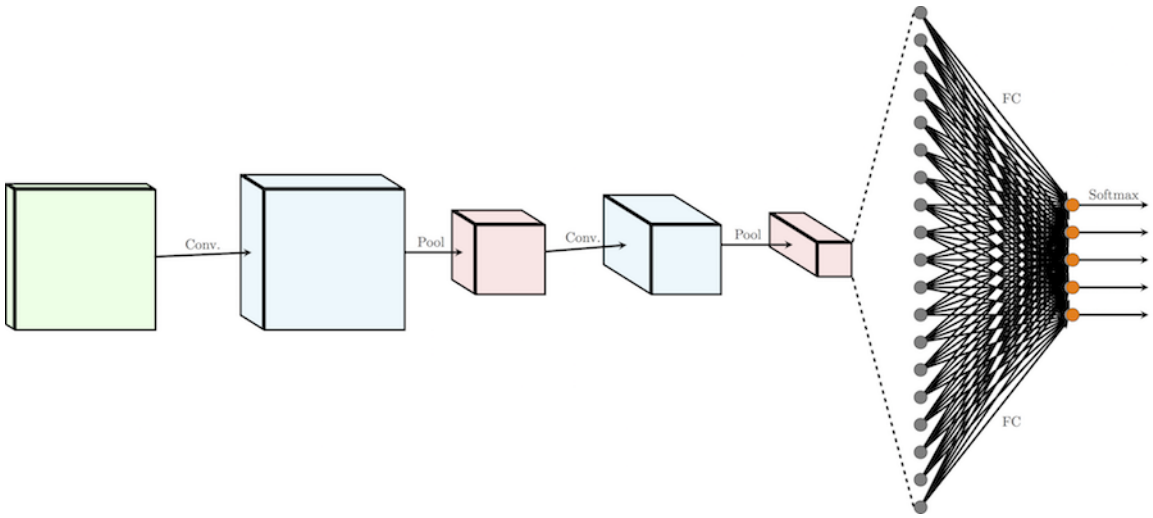

Figure 3.7: Example of a complex CNN with a fully connected layer at the end and 5 classes to predict.

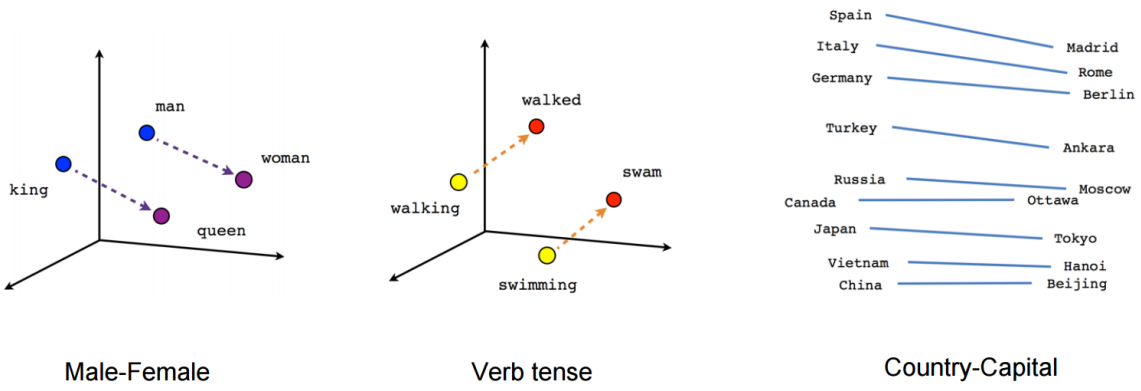

Figure 3.8: Examples of words representation in the space according to their word embedding values.

3.3 Word Embedding

A word embedding system turns words into vectors. It is a learned representation where words with similar meaning or belonging to the same topic have small distances between them. Word embeddings are represented as matrices, where each row belongs to a different word. These systems build vocabularies assigning index numbers to the terms appearing in it. The number of rows vary with the vocabulary size, but the number of columns is immutable. The words might be comparable between them, so the vector size for each one of the words have to be the same.

Embeddings give a semantic representation of the words with a numerical value. Fig. 3.8 [3] shows that the related words are close in space. For example, king and man are placed next to each other, but they keep the same relationship with woman and queen. Variations on verbs are equally separated, but past or adverbial forms trend to be together.

[3] https://www.tensorflow.org/tutorials/word2vec

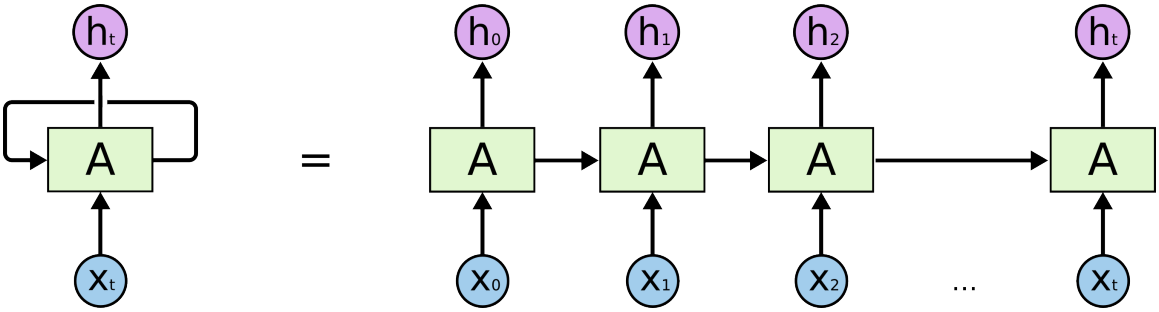

Figure 3.9: Unrolled Recurrent Neural Network where the inputs are the different time steps.

3.4 Recurrent Neural Network & LSTM

Traditional neural networks do not take time in consideration. There is only one state and it is the input to the system. Nevertheless, a lot of problems can be treated as sequences of data. Recurrent Neural Networks (RNN) [16] appeared to solve this issue. They have loops inside to store and process the previous information. Fig. 3.9 shows an unrolled RNN, which is a combination of neural networks passing the information from one neuron to the next one. Sometimes, we do not need to consider all the sequence information. In some cases we just only need a previous number of steps to do a prediction. But, in other cases we have long-term dependencies. In these cases the present prediction task has a relation with an input that was introduced far in the past. The long-term dependencies are solved with the Long Short Term Memory Networks (LSTM) [12], which are really useful in natural language processing problems, where this issues happens.

LSTM is usually used to perform tasks of Natural Language Processing (NLP), where the input is a text sequence. Fig. 3.10 [4] is an example of a predictive system for answering questions or a chat-bot. This shows the basics steps for NLP problems. The words are converted into vectors using a word embedding and introduced into

[4] https://ai.googleblog.com/2016/05/chat-smarter-with-allo.html

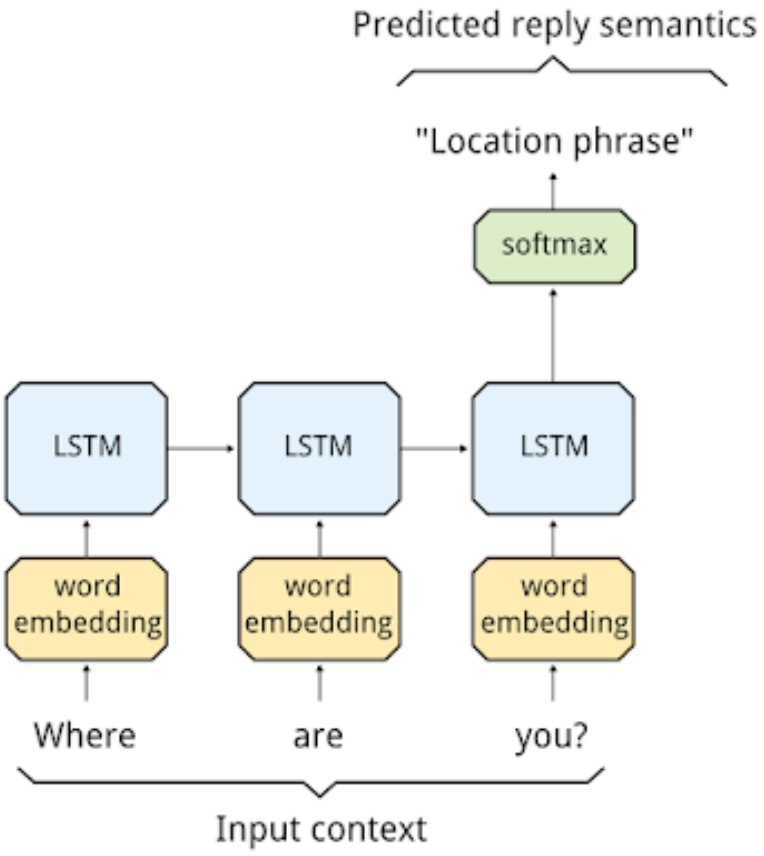

Figure 3.10: NLP example for question-answer systems.

the LSTM network.

Anyone interested in learning more about the RNN and LSTM, the blog Understanding LSTM Networks [5] provides a good introduction to this topic.

3.5 Image-based Food Menu Recognition: Our Model

In this section we introduce our proposed model. Fig. 3.11 shows an example of the structure of the model. The proposed system is based on image retrieval models. It means that it gives an output value based on the similarity between each tuple of

[5] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

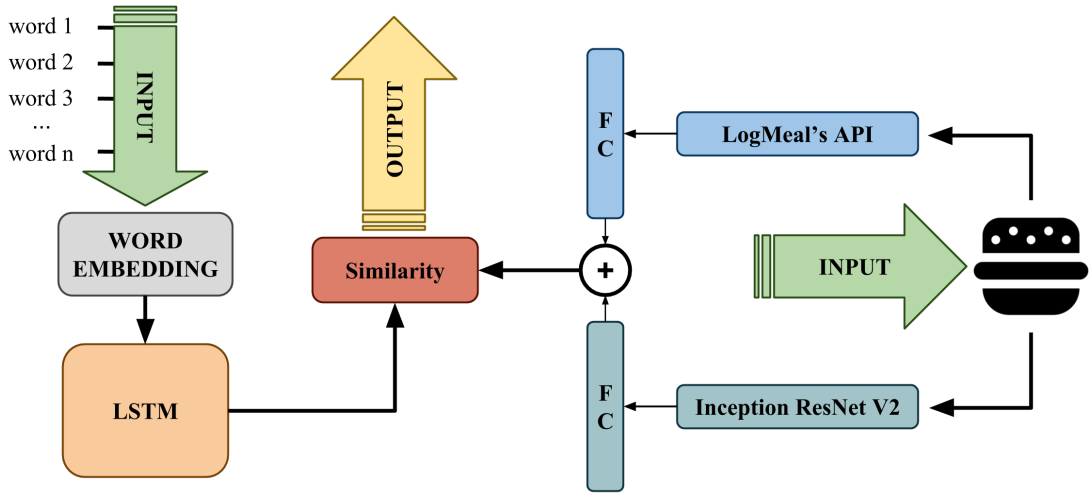

Figure 3.11: Image-based food menu recognition model. On one hand, the system gets an image and applies two different CNNs to generate the feature vectors. Each one is connected to a different fully connected layer to generate comparable structures and are combined performing an addition. On the other hand, the text sequence is processed by a word embedding and an LSTM. Finally, we compute the similarity between the two inputs using the euclidean similarity.

image and dish name provided. The prediction process will consist on running the predictive model for each menu item and the same meal picture. The response of the system will be the menu item with a higher similarity value.

3.5.1 Inputs

As we said in the introduction of this chapter, the result of our algorithm is a ranked list of the similarities given by two inputs: an image and a menu item. It means that the predictive model takes an image and a text sequence as input.

The image is converted in two vectors, which are the real input of the system. We use two different pre-trained CNNs to generate these vectors, which will not be trained but used as inputs to our system instead.

The first vector of the model is built using the response of the LogMeal's API [6]. The

[6] http://www.logmeal.ml

API output is composed from three different CNNs that predict the food type, food family [1] and the ingredients detected in the image [5]. The current response of the LogMeal's API is a classification of the image in a group of 11 family groups, 200 dishes and 1.092 ingredients. We are not using the ingredient classification because of the large dimensionality of the the output and the noise that this group introduces to the system. Finally, we concatenate the probabilities vector of the family group and the dish prediction.

The second vector uses the InceptionResNetV2 [25] model and it is generated from the results of the penultimate layer, composed by 1536 values. This CNN is pre-built in the Keras [9] framework and trained using the ImageNet [10] dataset. The main difference between this model and LogMeal's API is that LogeMeals was trained using only food images. However, this one have been trained using ImageNet [10], a generic dataset of pictures.

The text sequence input, representing the meals name, is encoded using a word embedding. The system assigns a unique identifier to each word, which is its reference to the rows embedding matrix. The inputs of our dataset are, in most of the cases, in English or Spanish. For this reason, we need a word to vector system supporting multiple languages. This is why we chose ConceptNet [24]. ConceptNet has a module named ConceptNet Numberbatch built for this specific purpose, which provides us a set of pre-trained vectors for the vocabulary in it. The words that don't appear in the ConceptNet vocabulary are initialized using a vector of random values. This kind of initialization is not a key factor because the vocabulary vectors are learned end-to-end in the model.

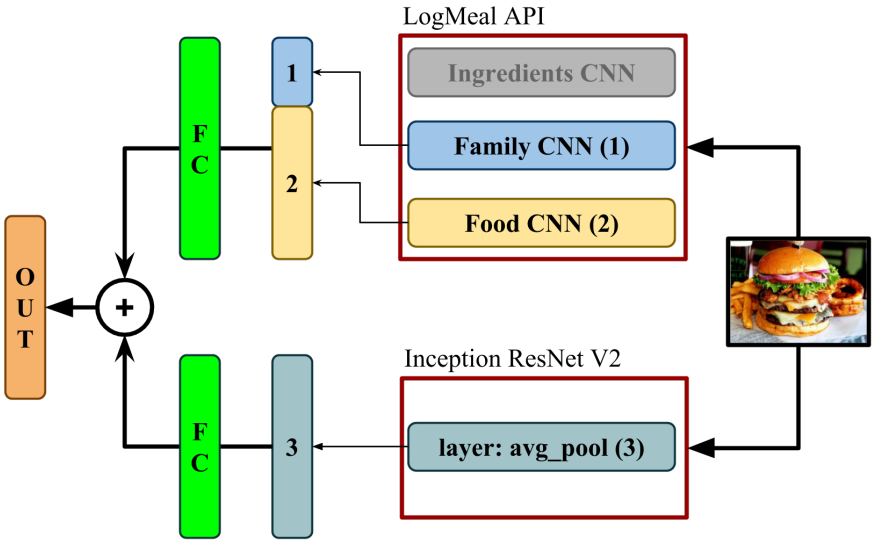

Figure 3.12: Image processing step. The system uses the food family and food recognition outputs of the LogMeal's API to create a new vector and connect it to a fully connected layer. The penultimate layer of the Inception ResNetV2 CNN is the feature vector which is connected to another FC. Finnally, both partial results are combined performing an addition.

3.5.2 Structure

This section explains the internal structure of the model shown in the fig. 3.11. The features vectors generated from the image (Fig. 3.12), one of them coming from the API response and the other from the CNN, are linked to a fully connected layer of 300 neurons.

This layer transforms the feature vectors to the same size, so we can combine them applying an addition operation. This process generates the first input to the similarity function. The second input of the similarity function comes from the text sequence of the meals name. It is generated using an LSTM network (Fig. X) with a shape of 300x1, so it is comparable with the vector built on the right side of the model 3.13. In the previous section we introduced that the input for the LSTM comes from a word embedding system. But the word embedding is not fixed, the values are mod-

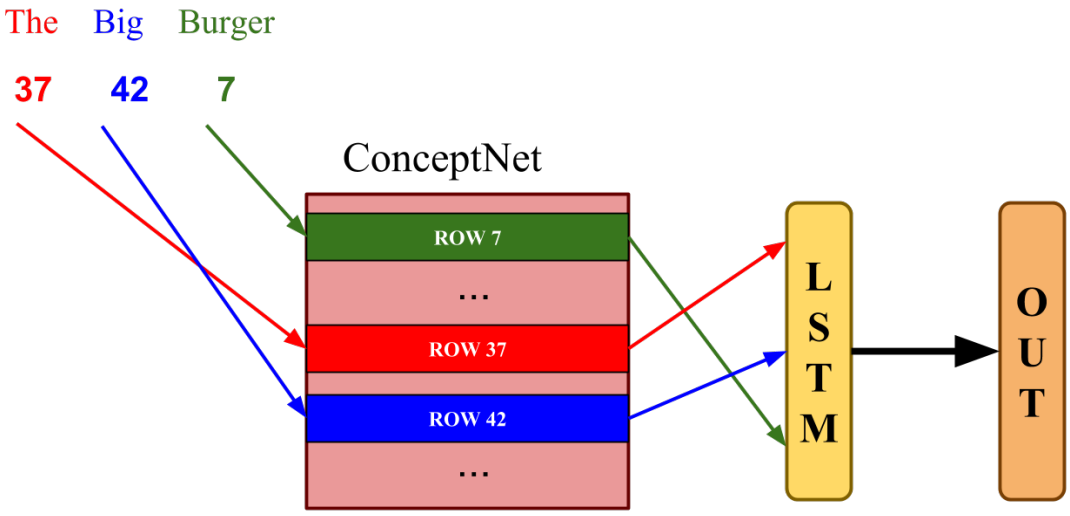

Figure 3.13: The text sequence is encoded using the ConceptNet word embedding. Which is connected to a LSTM generating the output vector. The model is trained end-to-end.

ified during the training. The words that are not present in the embedding matrix are initialized with random values.

3.5.3 Output

The output of the model is a value between 0 and 1. This number indicates the similarity between the two inputs of the system. If the output is close to 1 means that the image and the text are the same dish. Nevertheless, the result of our system is not a single similarity value. The response should be a ranking of the dishes in the menu sorted by their similarity to a certain image. It means that we need to run the model for each item in the menu on the same picture.

The similarity function used to build the algorithm is an adaptation of the euclidean distance 3.6.

$$\frac{1}{1 + \|q - p\|} \tag{3.6}$$

Chapter 4

Dataset

In this section we will introduce the dataset that we collected. Throughout this chapter we will talk about how the dataset was obtained, the software and difficulties that we faced, how it is structured and how did we split the dataset for training our algorithm.

4.1 Dataset collection

The dataset presented in this thesis and used for experimentation was built on our own using Yelp as the source of the information. Unfortunately, Yelp does not have any API or easy way to access to restaurants information. We needed to build a web scraper program to go through the multiple links that compose the information of a restaurant and save all the information in a easy-to-read format. The web-spider that we built is able to find all the restaurants from a query search URL. The spider starts going into every restaurant, gets the basic information, detects if the restaurant has a public menu, and scraps its content getting access to all the pictures uploaded by the users for each dish listed in the available menus. Fig. 4.1 shows the topology of

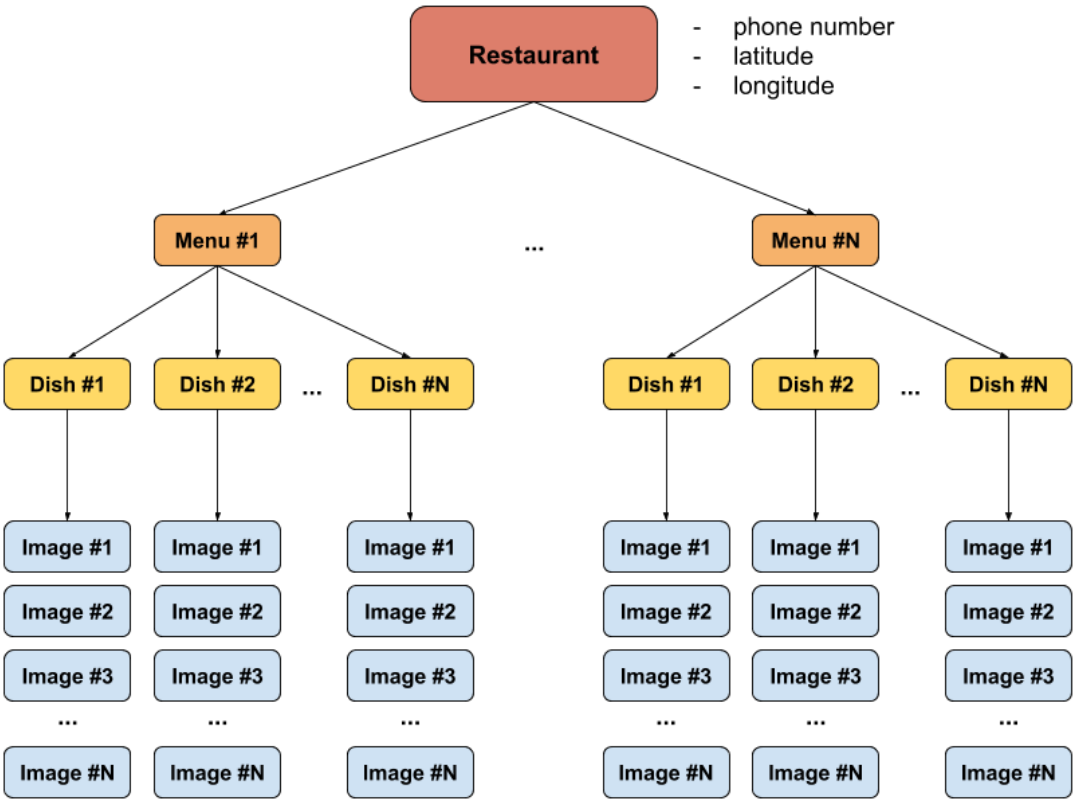

Figure 4.1: Yelp website information structure.

the restaurants information on the website.

4.2 Dataset characteristics

The dataset was built from restaurants located in California. We chose this location because of the amount of active Yelp users in this area. We make the dataset publicly available[1]. Anyone interested in building their own dataset can use the code provided. You would notice that there are two different projects. This section will only use YelpSpiders. The file YelpSpiders - algorithms - spider.py only need a Yelps URL search to run the web scraper and start collecting the information.

[1]https://goo.gl/EaUh4p

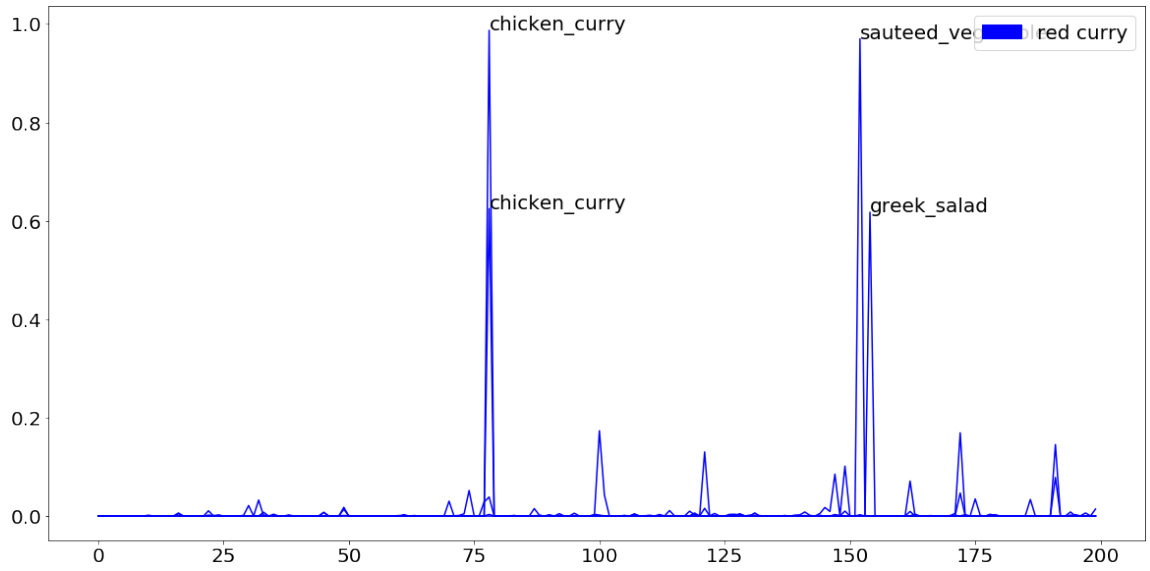

Figure 4.2: Red Curry food recognition LogMeal's API response.

Analyzing the response of the LofMeal's API, we decided to remove the ingredients information. It is appreciable, in the fig. 4.3 and 4.2, that the outputs for the same class have similar activation points, but they are different for images that represent different meals. Nevertheless, the ingredients recognition 4.4 is noisy and does not give a lot of relevant information. These features increase the dimensionality of the input, but the results are not better.

Table 4.2 shows the number of images, dishes and restaurants in the dataset. The fig. 4.5 is an histogram of the number of dishes per menu. Due to the location of the restaurants, there is a high probability of finding dishes in both english and spanish. This in fact introduces a problem: special characters. We encoded the text using the UTF-8 format, but there are some cases where the characters were represented by an empty symbol (_). In these example we decided to remove them from our dataset, because it was impossible to determine the missing character.

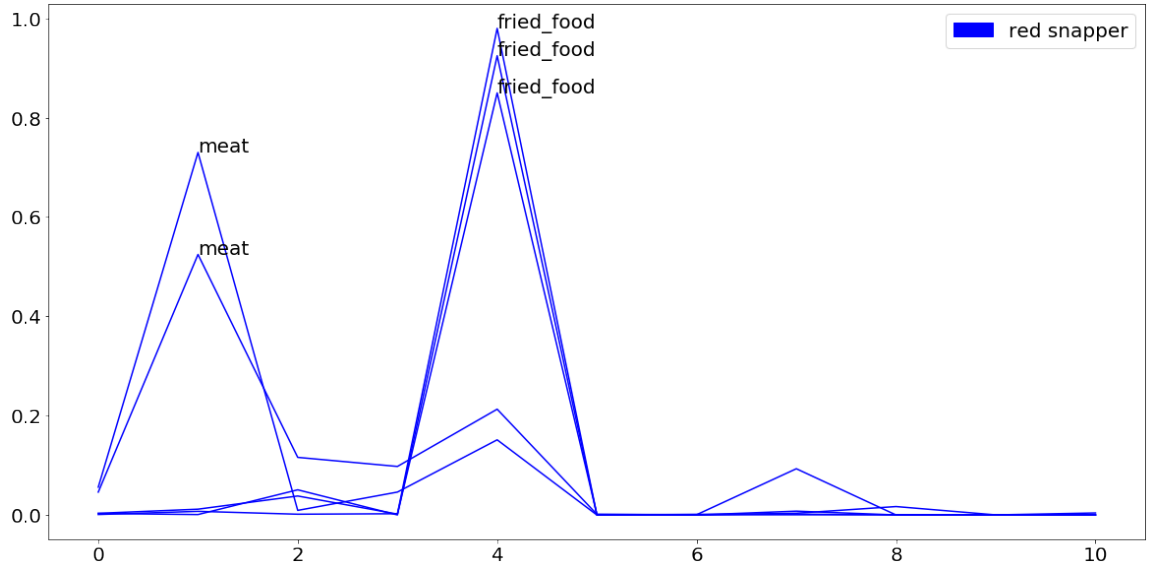

Figure 4.3: Red Snapper food family recognition LogMeal's API response.

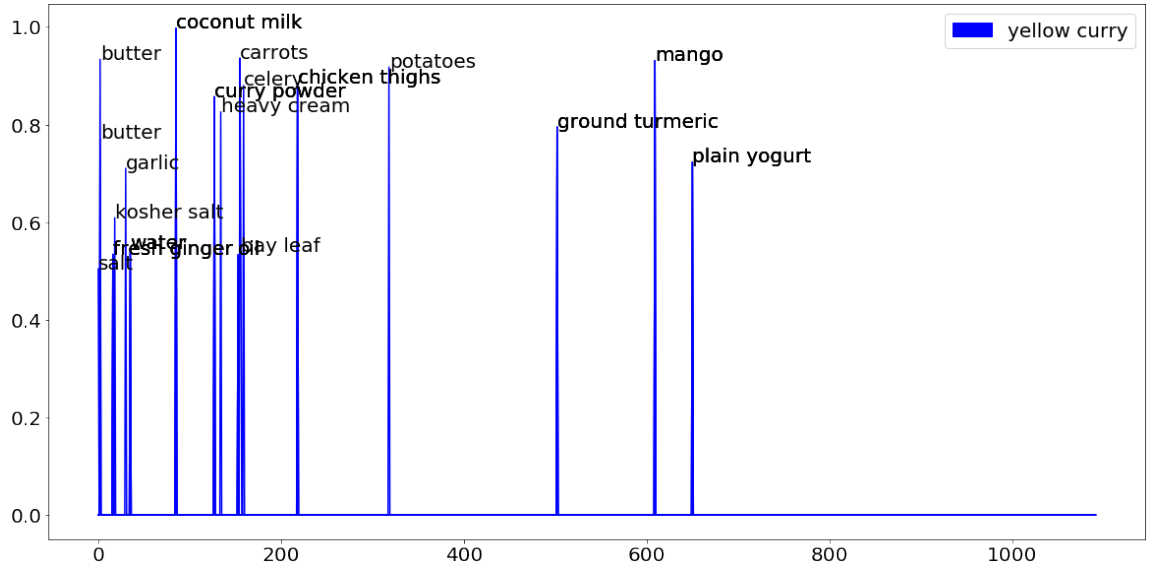

Figure 4.4: Yellow Curry food ingredients recognition LogMeal's API response. It is appreciable that the activation points are different, considering that 5 images are displayed at the same figure. Nevertheless, the food and family recognition share activation points between the images.

Table 4.1: Number of images, dishes and restaurants of the dataset.

Split	# of samples
# of images	53,877
# of dishes	3,498
# of restaurants	313

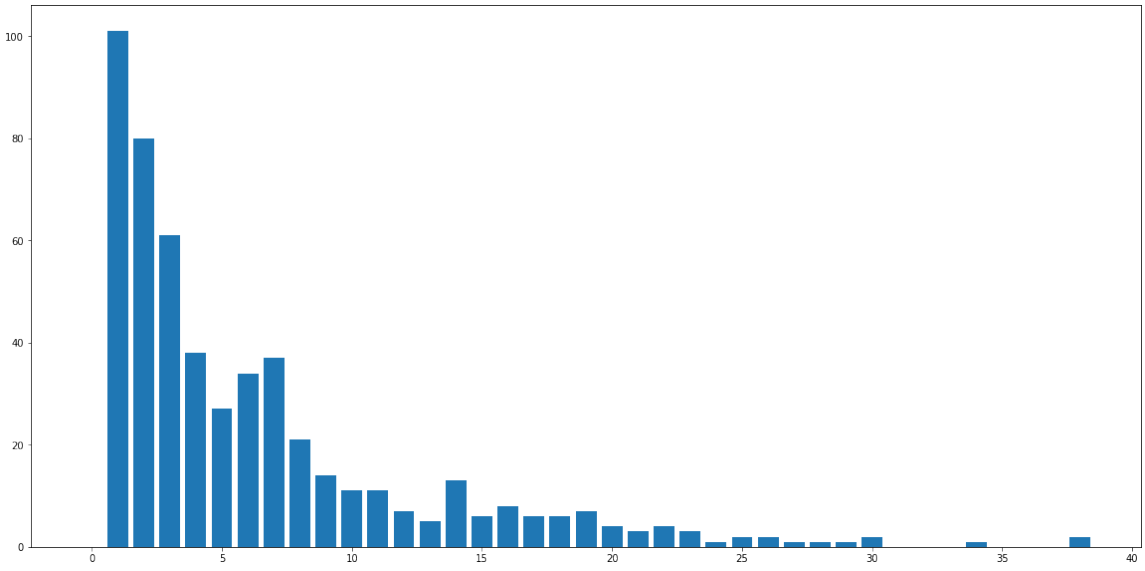

Figure 4.5: Histogram of the number of dishes per menu at X and the number of menus at Y.

The dataset provided in this work includes all the information and files needed to run the model, but the ones interested in a lighter version is also available in Google Drive [2]. The complete dataset follows the hierarchy shown in the fig. 4.6. The info.json file contains all the restaurants information, including all the images links to be downloaded. Each menu is located in a separate folder and inside them there are folders for each one of the dishes. The menu meals could have multiple images, all the images having two different files associated. The *.npy files are the vectors given by the LogMeal's API and the *˙cnn.npy are the features vectors extracted from the CNN. These two additional files share the name with the JPG picture to be easily related. The lighter dataset only contains the JSON files, there is no additional information included. Therefore, the reader can build the whole structure with the

[2]https://goo.gl/EaUh4p

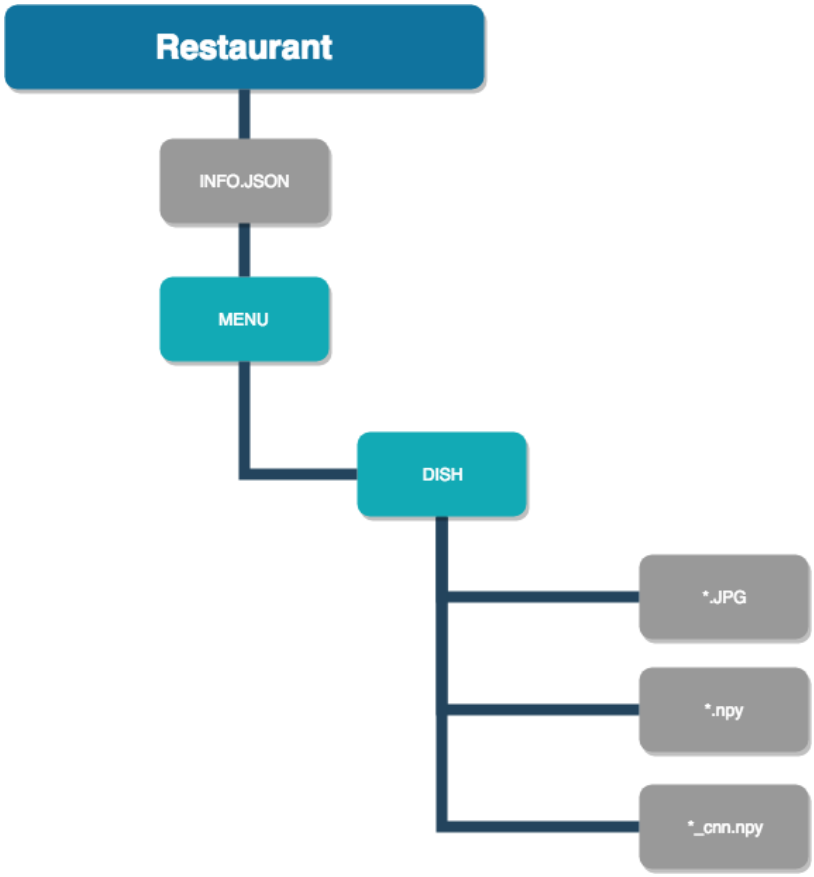

Figure 4.6: Tree schema representing the location of each one of the files and information in the dataset.

code provided in this work (YelpSpiders - algorithms - downloader.py).

4.3 Dataset Split

The dataset is split in three groups: training, validation and testing. Previously to the split process, we cleaned the data. This meant removing the dishes encoded in a not valid format or the ones that do not have more than 5 images. The dishes are randomly split in the three groups 4.7. The training group contains the 80% of the dishes, the 8% is included in validation and 12% of the meals are in the testing split.

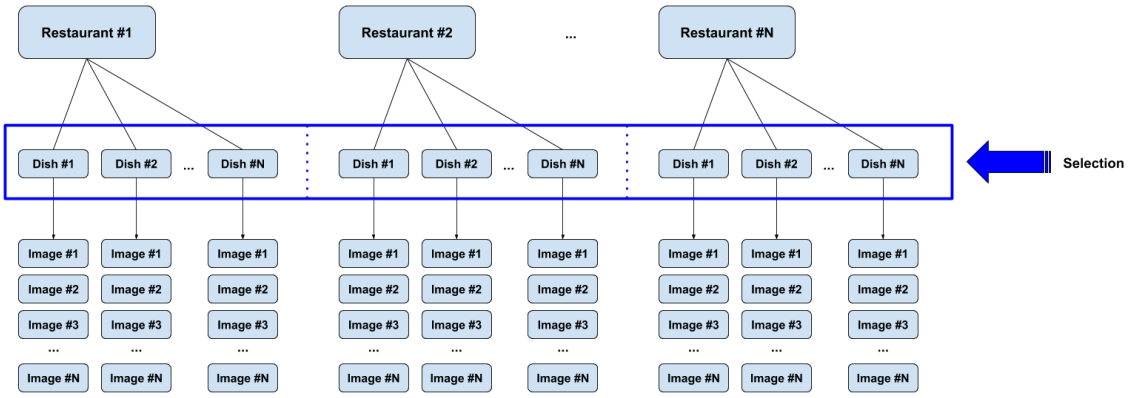

Figure 4.7: The dataset is separated in training, validation and testing performing a random selection of the dishes.

Table 4.2: The results presented in this thesis use the random split appearing in this table.

Split	# of images
Training	37,956
Validation	7,721
Test	10,794

The number of images of the groups are shown in Table 4.2.

After the selection of splits, we need to shape our data in an appropriate way for our problem. We have to consider that the training and evaluation are performed differently. The information included in the training group has been created with 50% of positive samples, where the similarity between the image and the text is 1, and another 50% of negative samples (similarity equals 0). The introduction of the negatives samples is made to avoid that the system learns only to generate the same value no matter the features provided. The validation and test splits are built comparing each image to a random selection of dishes. The menus size varies between 10 and 20 dishes randomly selected. We generate a random list instead of using the menus of the restaurants, to avoid food places that have few dishes in their menus.

We decided to create our own dataset to face a real problem. The available datasets in the Internet are usually standardized and cleaned of wrong examples, and they do not include noise in the data. Given the characteristics of our dataset, we can be sure that are as close as possible to images and names that we will find in a real environment.

Anyone interested in rebuilding the dataset split should run the script FoodMenuRecognition - algorithms - dataset.py in a first place. This script produces the training, val and testing splits. But the menus should be generated by running FoodMenuRecognition - utils - prepare˙data.py.

Chapter 5

Results

In this chapter we present the results obtained in our work, we introduce the loss and accuracy metrics used to evaluate the system and we show the set of experiments created to find the best combination of parameters to the problem that we are approaching.

5.1 Ranking Loss & Accuracy Top-1 Distance

The purpose of this section is to explain the error metric chosen to evaluate our system in the validation and test splits. This value has been used to choose the best parameters for our predictive model. Additionally, we propose a new and complementary accuracy metric for ranking evaluation.

In order to compare the performance of the different methods, we use the Ranking Loss [26]. Moreover, it is implemented in the scientific Python Library Scikit-Learn [20]. The implementation of the ranking loss error is described in the documentation of the scikit-learn framework, but the equation 5.1 shows the formula used to compute this value. That error metric does not only indicate if the response of the

system is right or wrong, it also gives a number about how far the prediction was wrongly ranked. A 0 ranking loss means that the system ranked the input at the right position, a 1 means that the prediction rank was the opposite of the expected one (the lower the better).

$$\text{ranking loss}(y, f) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \frac{1}{|y_i|\left(n_{labels} - |y_i|\right)} \{(k, l) : f_{ik} < f_{il}, y_{ik} = 1, y_{il} = 0\} \tag{5.1}$$

To complement the ranking loss error metric, we introduce our own accuracy metric in Eq. 5.2, which we call accuracy top-1 distance. This measure evaluates how close the ranked result is to the top, normalized between 0 and 1. The difference with the ranking loss is that our metric only takes in consideration the distance from the position of the predicted class to the top of the ranking. We normalize the output between 0 and using the number of labels in our ranking.

$$\text{accuracy top-1 distance} = \frac{n_{labels} - 1 - ranking_{position}}{n_{labels} - 1} \tag{5.2}$$

5.2 Experimental setup

The selection of the best values combination was done using a forward propagation-grid search, Table 5.1 shows the results. The configuration of the network was fixed at the first iterations. For each step in the grid search we select the value retrieving the best error at the testing group.

Anyone interested in running their own version of the model can use the software provided with this document. You will notice that there are two different projects. This section will only use FoodMenuRecognition. The file FoodMenuRecognition - algorithms - model.py is the one containing all the functions to train and evaluate the system.

The following sections will explain each one of the components of the model configuration displayed at the table. The results of the table (ranking loss and accuracy top-1 distance) are calculated training the system 5 times during 10 epochs. The values are the median over the best epoch of each iteration.

The best results were obtained at the first epoch with a batch size of 64 samples and without applying any data augmentation or normalization process.
Following, we detail the different model variants that we compare in the experimental section:

5.2.1 Similarity

We introduced two similarity function candidates. The euclidean similarity (euclidean) is based on the euclidean distance, it has been modified to just return values between 0 and 1. The Pearson similarity (pearson) 5.3 is the absolute value of the Pearson correlation. Using the absolute value we get values between 0 and 1. Additionally, we dont need to know if the relation between the vectors is positive or negative, just if exists a relationship.

$$\rho = \left| \frac{\text{cov}(X, Y)}{\sigma_x \sigma_y} \right| \tag{5.3}$$

5.2.2 Loss optimizer

The optimizer should compare the true value with the predicted one, which comes from the similarity function. The range of the values are between 0 and 1, so we chose two loss functions expecting this range of outputs. The binary cross-entropy (BCE) [23] computes the entropy between the two probabilities distributions. It is a common used loss function for the binary classification problems. The contrastive loss (CL)

[11] is a loss function created to the siamese text problems. It is a distance-based system and tries to minimize the separation between examples of the same semantic topic.

5.2.3 CNN

The CNN field at the table can take 3 different values. It defines the feature vectors to train and evaluate the model. *LM* means that the model is trained using only the features from the LogMeal API. *LM+Inc* in the parameter indicates that the model uses the combination of the vectors from the API and the Inception ResNet V2 CNN, as we said at the section 3.5 when we explained our proposed model. Finally, when the field is *Inc*, the model only uses the Inception vector.

5.2.4 Sample Weight

The last parameter to select is the sample weight. It indicates whether we want to assign a weight value to each dish in relation with the amount of images that it contains compared with the total. This kind of weightings are useful when the datasets are unbalanced, it gives more importance to the samples that are less frequent.

Table 5.1 shows the results of the grid search applied. The last row of the table is the baseline error (based on a random selection of an item in the menu) and accuracy value over validation and test. We have to considerate that the values of the ranking loss follow the rule, the lower the better. Meanwhile, the accuracy has the opposite behavior, we want to achieve the higher possible value. The policy we follow to choose the best parameter uses the ranking loss error over the test.

Table 5.1: Grid Search Results. The measure is the similarity function to evaluate (Euclidean or Pearson). The loss column select the best optimization function (binary cross-entropy or contrastive loss). CNN type indicates the combination of CNNs used in the model (LogMeal's API and Inception ResNetV2). The weight column indicates if the systems is using sample weight or not. The last two groups of columns show the results of the models using the groups of validation and testing. The ranking loss (r.loss) wants to achieve the lower possible value. Meanwhile, the accuracy top-1 distance (acc.) pursues the opposite objective. The best configuration of the system is shown at the last row with the baseline values for this problem.

				val		**test**	
measure	**loss**	**CNN type**	**weight**	**r. loss**	**acc.**	**r. loss**	**acc.**
euclidean	binary	LM	NO	0.384	0.623	**0.362**	0.671
pearson	binary	LM	NO	0.416	0.602	0.395	0.639
euclidean	binary	LM	NO	0.384	0.623	**0.362**	0.671
euclidean	contrastive	LM	NO	0.405	0.398	0.375	0.664
euclidean	**binary**	LM	NO	0.384	0.623	0.362	0.671
euclidean	**binary**	LM+Inc	NO	0.372	0.641	**0.351**	0.678
euclidean	**binary**	Inc	NO	0.443	0.572	0.413	0.598
euclidean	**binary**	**LM+Inc**	NO	0.372	0.641	**0.351**	0.678
euclidean	**binary**	**LM+Inc**	YES	0.396	0.612	0.378	0.668
euclidean	**binary**	**LM+Inc**	**NO**	**0.372**	**0.641**	**0.351**	**0.678**
baseline				0.5	0.5	0.5	0.5

The first two rows of the table evaluate the two similarity measures (Pearson and Euclidean Similarity). Both similarity measures are tested with the same loss optimizer, CNN and sample weight values to be comparable. The error of the Euclidean similarity is 0.033 points better than the one using the Pearson function. That means that the following iterations of the search use the first one.

The rows from 2 to 3 of the table look for the better loss optimization function over two possible options, the binary cross-entropy and the contrastive loss. The function selected at the first place is the one retrieving a lower ranking loss error.

The following three rows evaluates the different image representation. If we compare the two CNN, LogMeal and Inception, the fist one works better. It is because the LogMeal CNN is trained using food images. Despite this considerations, the best results are got by the model using the combination of the two CNN. The both network complement each other, getting better results when using them.
Finally, the next parameter to evaluate is the sample weight. The difference of not using or using the sample weight is of 0.027 points. The dishes names are not equally distributed across the restaurants, some of them are more popular and are shared in a lot of places. Because of that is not a good practice, in this case, to give the same weight to all the examples.

Concluding the table analysis, the best combination of parameters for our model improves the baseline by a 15%. The best ranking loss for the test group is 0.351 and the accuracy top-1 distance is 0.678. It means an improvement of 0.149 and 0.178 points respectively over the baseline.

5.3 Visual Results Analysis

In this section we show some visualizations of the results. The visualization contains a picture of the meal, the ranked results of our system and the true prediction for the image. Additionally, the titles of the figures have the error and accuracy for each one of the samples.

Figs. 5.1, 5.2, 5.3, 5.4, 5.6a nd 5.8 show that the cases where the system works better is when the picture present a single piece of food and the image is clear and centered. The fig. 5.9 is an exception of the previous premise, because the accuracy is very high but the image does not present the best conditions to be recognized. Figs. 5.7, 5.10, 5.11 and 5.12 are examples of bad images getting bad results. These images contain multiple meals on them, making the recognition harder. Fig. 5.5 is another exception, the picture contains an unique food meal on it, but the ranking loss is 0.57.

Additionally, it is appreciable in the figures of this section that the dishes with long names are usually at the bottom of the ranking. It is because these meals do not contain a lot of images and are not very popular in the restaurants. So, the model is not able to learn them and retrieve good predictions.

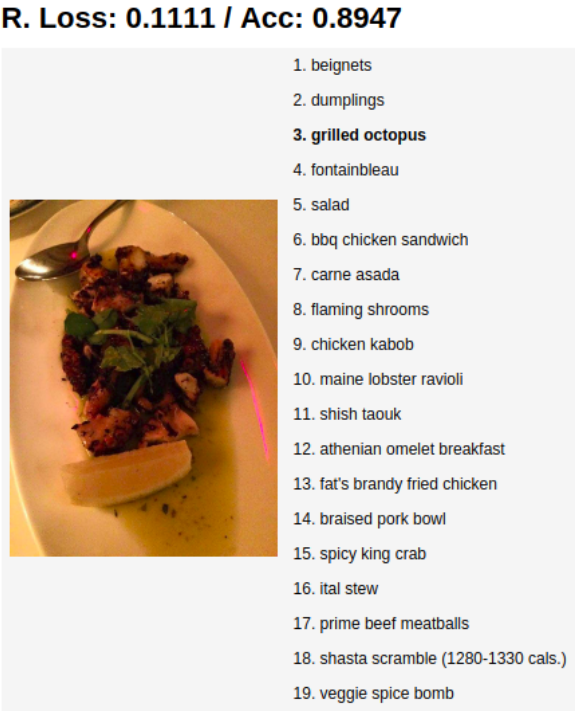

Figure 5.1: Grilled octopus.

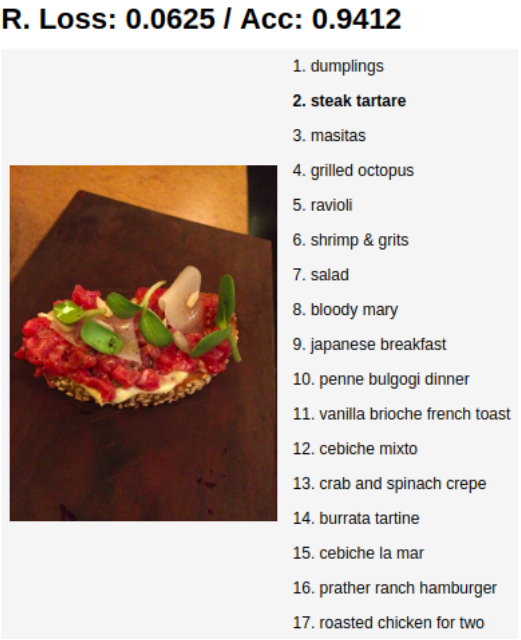

Figure 5.2: Steak tartare.

Figure 5.3: Ravioli.

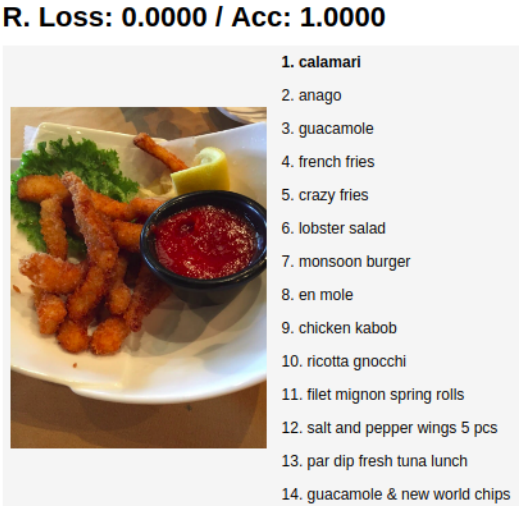

Figure 5.4: Calamari.

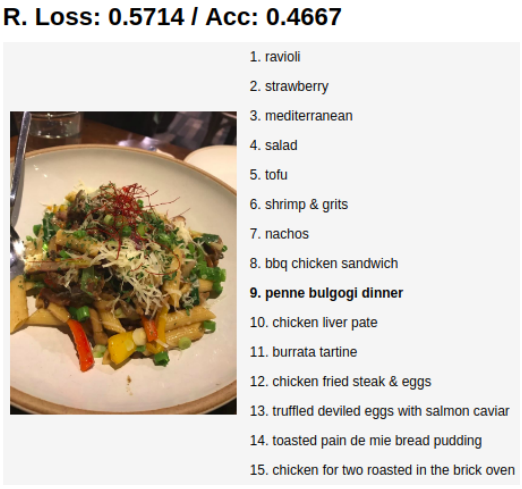

Figure 5.5: Penne buigogi dinner.

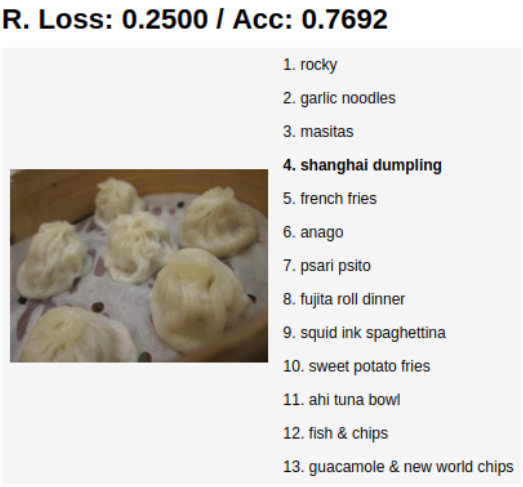

Figure 5.6: Shangai dumpling.

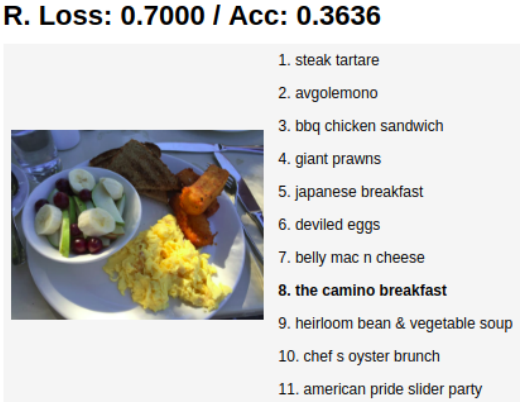

Figure 5.7: The camino breakfast.

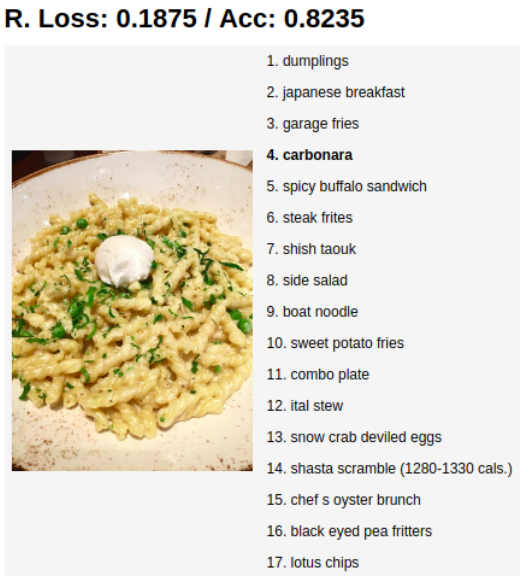

Figure 5.8: Carbonara.

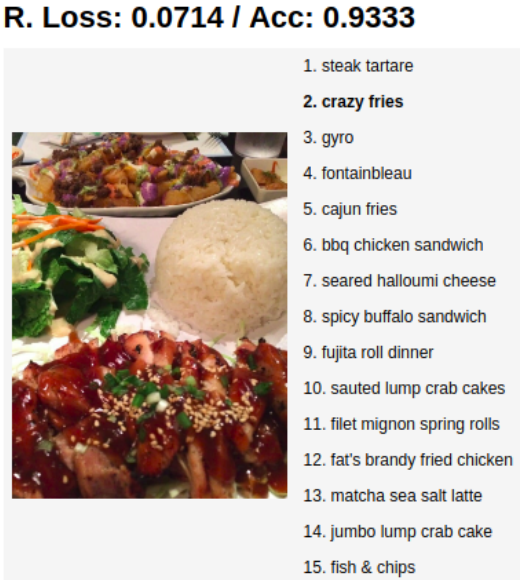

Figure 5.9: Crazy fries.

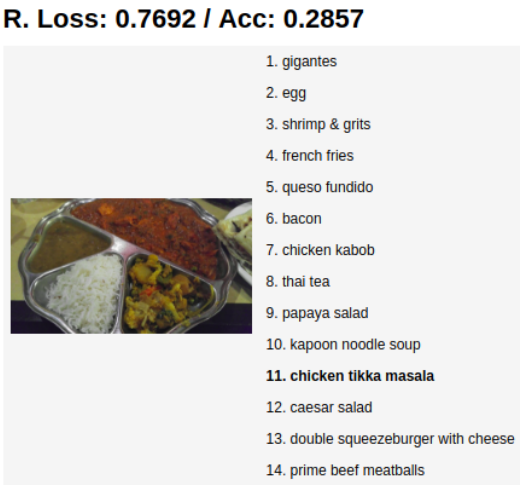

Figure 5.10: Chiken tikka masala.

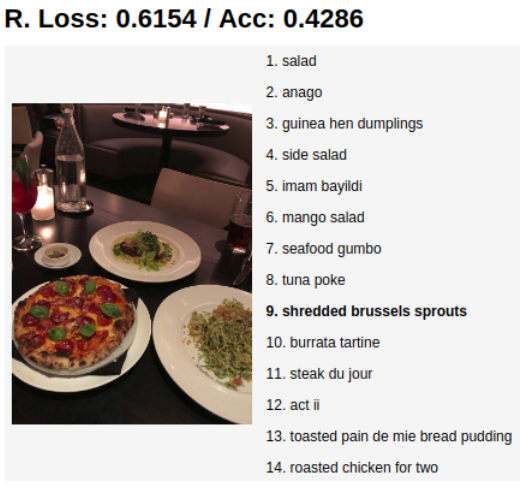

Figure 5.11: Shredded brussels sprouts.

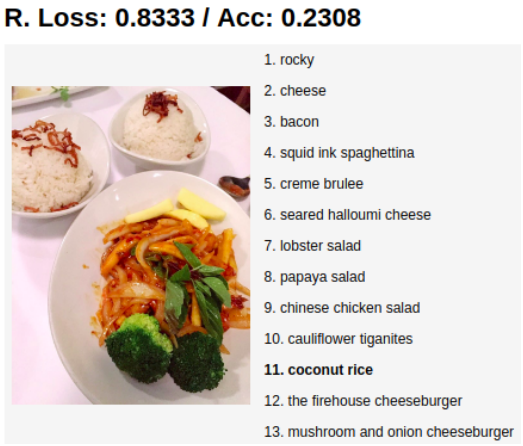

Figure 5.12: Coconut rice.

Chapter 6

Discussion

In this chapter we discuss the final results of the project. We give an explanation to the results in relation with the difficulty of the problem and the results shown in the previous chapter. Additionally, we expose the challenge of building a new dataset for the project. We present the reasons why this could be a good or a bad idea.

The best results were accomplished by the input that combines the responses from the LogMeal's API and the Inception ResNet V2 CNN. Despite the fact that the LogMeal vector works better by itself compare with the latter, they complement each other, and the combination performs better than either of them.
The previous chapter section, Visualize Results 5.3, shows some predictions done by our system for a particular image. The following lines are focused on discuss the kind of images and the problems that they present, the names of the dishes and the cases where the system works better.

The images of the dataset are taken and tagged by Yelps users. It means that the pictures uploaded to the site are not verified and could be wrong. It is not really common to find images misclassified, but we have found a lot of them that contains

multiple dishes in a single picture or where the food is not the key-factor in the image. Most of the meals do not have a single food type appearing in it, because they contain several pictures. The users take photos of their dishes including context information, and it is a possibility that this information includes other peoples meals. This makes more difficult to classify the sample.

The main difficulty for the algorithm is dealing with a high variety of names. The restaurants have some speciality dishes that they name at their own. These meals are really difficult to classify, even for a human. Visualizing the results and analyzing the responses of a random selection of the predictions, we have found some properties that usually work better in our system. The meals that contain common food names tend to get better results than the ones with exotic names. This fact is due to two main reasons: the first one is that the dataset has a lot of examples with common names and can learn them better, and the second one is that the exotic names do not tend to appear at the word embedding matrix, so the system has no initial information of them. Moreover, these names are present in just a few restaurants, so the system does not have enough examples to learn.

Chapter 7

Conclusions and Future Work

In this chapter we introduce the conclusions of our work, determining the key points and the importance of it to the food recognition field. Additionally, we propose some future work that could improve our current results and new applications of our work.

7.1 Conclusions

In this section we list the contributions that we have done to the scientific community.

- We present a new dataset composed by the dishes and images of the restaurant's menu. The dataset contains 53,877 images, 3,498 dishes and 313 restaurants. It is public and open to everybody. We explain how we built it and provided the code to modify or expand the dataset.

- We introduce the use of semantic information by means of LogMeal's API to perform dish recognition.

- We propose a new model that determines the similarity between a food image and a menu item of a restaurant. We run a set of experiments to determine the best parameters of our model, using the introduced ranking loss and our ranking accuracy metric. We compared the obtained results to the baseline of 0.5, where we improve a 15% to the baseline.

7.2 Future Work

The research done in this thesis was focused on recognizing a picture of a meal from a menu's list. The future work planned to apply this model introduces the GPS information of the images. The location of the user gives us a list of two or three candidate restaurants where they are eating. Combining the menus of these restaurants and applying the proposed system we would be able to determine where and what a person is eating.

References

[1] Eduardo Aguilar, Marc Bolaños, and Petia Radeva. Food recognition using fusion of classifiers based on cnns. In *International Conference on Image Analysis and Processing*, pages 213–224. Springer, 2017.

[2] Eduardo Aguilar, Beatriz Remeseiro, Marc Bolaños, and Petia Radeva. Grab, pay and eat: Semantic food detection for smart restaurants. *arXiv preprint arXiv:1711.05128*, 2017.

[3] Vinay Bettadapura, Edison Thomaz, Aman Parnami, Gregory D Abowd, and Irfan Essa. Leveraging context to support automated food recognition in restaurants. In *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*, pages 580–587. IEEE, 2015.

[4] S Boesveldt, J Frasnelli, AR Gordon, and JN Lundström. The fish is bad: negative food odors elicit faster and more accurate reactions than other odors. *Biological psychology*, 84(2):313–317, 2010.

[5] Marc Bolaños, Aina Ferrà, and Petia Radeva. Food ingredients recognition through multi-label learning. In *International Conference on Image Analysis and Processing*, pages 394–402. Springer, 2017.

[6] Marc Bolanos and Petia Radeva. Simultaneous food localization and recognition. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 3140–3145. IEEE, 2016.

[7] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European Conference on Computer Vision*, pages 446–461. Springer, 2014.

[8] Gheorghe M Bota and Peter B Harrington. Direct detection of trimethylamine in meat food products using ion mobility spectrometry. *Talanta*, 68(3):629–635, 2006.

[9] François Chollet et al. Keras. https://keras.io, 2015.

[10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[11] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 1735–1742. IEEE, 2006.

[12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[13] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[14] Huan-Chung Li and Wei-Min Ko. Automated food ontology construction mechanism for diabetes diet care. In *Machine Learning and Cybernetics, 2007 International Conference on*, volume 5, pages 2953–2958. IEEE, 2007.

[15] Chang Liu, Yu Cao, Yan Luo, Guanling Chen, Vinod Vokkarane, and Yunsheng Ma. Deepfood: Deep learning-based food image recognition for computer-aided

dietary assessment. In *International Conference on Smart Homes and Health Telematics*, pages 37–48. Springer, 2016.

[16] LR Medsker and LC Jain. Recurrent neural networks. *Design and Applications*, 5, 2001.

[17] Jonas Mueller and Aditya Thyagarajan. Siamese recurrent architectures for learning sentence similarity. In *AAAI*, pages 2786–2792, 2016.

[18] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 689–696, 2011.

[19] World Health Organization et al. *Food and health in Europe: a new basis for action*. World Health Organization. Regional Office for Europe, 2004.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[21] Paul Rozin, Claude Fischler, Sumio Imada, Allison Sarubin, and Amy Wrzesniewski. Attitudes to food and the role of food in life in the usa, japan, flemish belgium and france: Possible implications for the diet–health debate. *Appetite*, 33(2):163–180, 1999.

[22] Amaia Salvador, Nicholas Hynes, Yusuf Aytar, Javier Marin, Ferda Ofli, Ingmar Weber, and Antonio Torralba. Learning cross-modal embeddings for cooking recipes and food images. *Training*, 720:619–508, 2017.

[23] John Shore and Rodney Johnson. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *IEEE Transactions on information theory*, 26(1):26–37, 1980.

[24] Robert Speer and Joanna Lowry-Duda. Conceptnet at semeval-2017 task 2: Extending word embeddings with multilingual relational knowledge. *arXiv preprint arXiv:1704.03560*, 2017.

[25] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.

[26] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer, 2009.

[27] Wen Wu and Jie Yang. Fast food recognition from videos of eating for calorie estimation. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pages 1210–1213. IEEE, 2009.

[28] Ruihan Xu, Luis Herranz, Shuqiang Jiang, Shuang Wang, Xinhang Song, and Ramesh Jain. Geolocalized modeling for dish recognition. *IEEE transactions on multimedia*, 17(8):1187–1199, 2015.

[29] Jun Yu, Xiaokang Yang, Fei Gao, and Dacheng Tao. Deep multimodal distance metric learning using click constraints for image ranking. *IEEE transactions on cybernetics*, 47(12):4014–4024, 2017.